UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

CAROLINA MOLINARI MONTEFORTE
LEANDRO DA CRUZ CIRQUEIRA
LEVY SALLES BISPO DE OLIVEIRA
WILLIAM SILVA VEÇOSO

Sistema de Recomendação de Filmes: Uma Abordagem de Aprendizado de Máquina

São Paulo, 2024

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Repositório de dados: <https://github.com/CirqueiraLeandro/CLLW-PROJETO-III**>**

1.1 Contexto

Na indústria cinematográfica, algoritmos de recomendação de filmes têm ganhado importância com a popularidade das plataformas de streaming e o volume de dados gerados nesses serviços. Com a quantidade de conteúdo produzida pelos apps de streaming e a variedade de títulos, encontrar conteúdo relevante com os interesses pessoais de cada um tornou-se um desafio para os usuários.

1.2. Motivação

Diante do contexto apresentado, entender o que torna um filme atraente para diferentes públicos é essencial. A motivação para o desenvolvimento de um sistema de recomendação de filmes surge da necessidade de proporcionar uma experiência personalizada e acessível para diferentes públicos, promovendo a inclusão digital e cultural, ao mesmo tempo em que facilita o acesso a conteúdos diversos e relevantes, contribuindo para a democratização do entretenimento e o engajamento da comunidade com tecnologias avançadas de recomendação

1.3 Justificativa

A criação de um sistema de recomendação de filmes justifica-se pelo impacto positivo na inclusão digital e no acesso democrático à cultura, promovendo a disseminação de conhecimento sobre tecnologias de recomendação e facilitando o acesso a conteúdos audiovisuais relevantes, beneficiando tanto usuários individuais quanto a comunidade em geral por meio de possíveis ações educativas e sociais.

Este trabalho utiliza o TMDB Movie Dataset, que inclui dados sobre o enredo, elenco, equipe técnica, orçamento e receitas de vários filmes. A base de dados foi gerada pelo TMDB API, e oferece uma oportunidade de desenvolver um sistema de recomendação de filmes que atendam aos gostos e expectativas dos usuários, justificando a criação deste sistema com a necessidade atual de otimizar a experiência de escolha e melhorar o engajamento com o conteúdo.

1.4 Objetivos

- Desenvolver um sistema de recomendação de filmes, para facilitar a escolha e o acesso a conteúdos audiovisuais relevantes para cada pessoa.
- Avaliar diferentes algoritmos de recomendação (colaborativos e baseados em conteúdo) para determinar o mais eficiente na recomendação de filmes.
- Identificar possíveis inconsistências nos dados, como dados nulos ou incorretos, e propor métodos de correção para garantir a precisão do sistema de recomendação.

2 REFERENCIAL TEÓRICO

Segundo Carlos A. Gomez-Uribe e Neil Hunt (2015), o sistema de recomendação da Netflix influência em 80% das horas assistidas no serviço de streaming. Além disso, segundo um estudo descrito pelo jornal The Wrap, em 2016, o tempo médio que o usuário leva para escolher um título da Netflix é quase o dobro do que um usuário de TV a cabo leva para escolher um canal. Esses fatos apresentados reforçam a justificativa em desenvolver tais sistemas.

Para construí-los, notamos que os sistemas de recomendação podem ser aplicados de diversas maneiras, não existindo uma solução única para ser utilizada em todas as situações. A escolha do método depende bastante do tipo de informações disponíveis. Ao falarmos de filmes, por exemplo, temos muitas informações disponíveis como título, ano, diretores, atores, gênero, avaliações de uma determinada rede de usuários e assim por diante. As avaliações podem ser explícitas (como “curtidas”, “gostei ou não gostei”, estrelas) ou implícitas (tempo assistido). As abordagens mais conhecidas são as de filtragem colaborativa e a baseada em conteúdo.

2.1 Filtragem colaborativa

A filtragem colaborativa foi a primeira abordagem a ser estudada para sistemas de recomendação, baseia-se na premissa de que usuários com gostos similares tendem a gostar dos mesmos itens. Dessa forma, analisa as interações entre usuários e itens (como filmes, produtos, etc.) para identificar padrões e fazer sugestões.

Apesar de ser bastante popular, a filtragem colaborativa pode enfrentar limitações quando há poucos dados disponíveis sobre um novo usuário ou item, dificultando a geração de recomendações precisas.

Existem duas principais abordagens para implementar a filtragem colaborativa: a baseada em memória e a baseada em modelos de fatores latentes. A primeira calcula a similaridade entre usuários ou itens para fazer recomendações, enquanto a segunda busca características latentes que explicam as preferências dos usuários. O KNN e a SVD são exemplos populares dessas abordagens, respectivamente.

2.2 Filtragem baseada em conteúdo

A filtragem baseada em conteúdo utiliza os atributos dos itens para fazer recomendações. Ou seja, ao invés de analisar o que usuários semelhantes gostaram, essa abordagem analisa o que o usuário já gosta e busca itens com características parecidas.

Essa técnica é bastante útil para lidar com o problema de falta de informações para novos usuários ou itens, pois não depende de um histórico de interações de outros usuários.

Há várias formas de implementar a filtragem baseada em conteúdo, como a utilização de árvores de decisão e técnicas de word embedding. As árvores de decisão constroem modelos preditivos que mapeiam as características dos itens às preferências do usuário, enquanto o word embedding transforma palavras em representações numéricas, permitindo analisar textos como sinopses de filmes e identificar similaridades.

2.3 Filtragem híbrida

Ainda podemos usar as duas técnicas em conjunto para gerar resultados ainda mais precisos, tentando evitar as limitações das duas técnicas acima citadas e complementar os resultados de maneira mais eficiente.
As abordagens híbridas tendem a ser mais utilizadas no mercado pois trazem melhores resultados.

Neste trabalho optamos por utilizar a filtragem colaborativa como primeira abordagem para os testes, mais especificamente o algoritmo SVD, dado que as bases disponíveis que escolhemos tem também suas limitações de conteúdo. A base de dados que trabalhamos nesse primeiro teste contém apenas dados de usuários, id do filme, nota e horário, e não há relação direta com a base de metadados do TMDB Movie Dataset, sendo inviável o enriquecimento dela com os dados dos filmes assistidos para tentar uma abordagem híbrida, por exemplo.

2.4 SVD

O método selecionado para esse projeto tem como base a técnica de fatoração de matrizes pela decomposição em valores singulares, ou, apenas, Singular Value Decomposition (SVD).

A SVD é uma técnica matemática que permite "desembaraçar" dados complexos em componentes mais simples. Diminuindo a quantidade de informações necessárias para representar os dados, tornando os cálculos mais eficientes.

Trata-se de uma técnica de fatoração matricial que reduz o número de características de um conjunto de dados. Essa fatoração permite a descoberta de fatores latentes que os filmes podem compartilhar de forma implícita.

A matriz de avaliações representa as notas que os usuários dão aos filmes, sendo decomposta em duas matrizes menores: uma de usuários e uma de filmes. Cada linha nas matrizes representa um "vetor latente", que corresponde a um conceito ou característica (por exemplo, gênero, ator favorito). Ao calcular o produto escalar entre os vetores latentes de um usuário e um filme, podemos prever a nota que o usuário daria para aquele filme.

A SVD permite que os sistemas de recomendação entendam as preferências dos usuários identificando os gêneros, atores ou outros fatores que os usuários mais apreciam, recomenda filmes relevantes que se encaixam no perfil de cada usuário, mesmo que eles não tenham visto filmes semelhantes antes, e pode lidar com dados esparsos, podendo utilizar matrizes com muitas entradas faltantes (quando um usuário não avaliou um filme), o que é comum em sistemas de recomendação.

Optamos por utilizar a biblioteca Surprise já que ela é uma biblioteca Python dedicada a algoritmos de predição e recomendação. Dessa maneira, a função matemática SVD já está disponível como função dentro da Surprise e avaliado por meio de validação cruzada com 5 divisões (folds). A validação cruzada foi realizada dividindo os dados em 5 partes. A cada iteração, o modelo foi treinado em 4 partes e testado na parte restante, de forma que cada subconjunto fosse usado tanto para treino quanto para teste ao longo do processo. Esse método fornece uma medida mais confiável da performance do modelo, uma vez que utiliza a totalidade dos dados.

2.5 Avaliação do modelo

Uma vez treinado o modelo, as métricas de avaliação a serem utilizadas serão:

- RMSE (Root Mean Squared Error): Calcula a raiz quadrada da média dos erros quadráticos entre os valores preditos e os valores reais. É sensível a outliers e penaliza grandes erros.
- MAE (Mean Absolute Error): Calcula a média dos valores absolutos dos erros. É menos sensível a outliers do que o RMSE.

Os resultados obtidos em cada um dos cinco folds foram os seguintes:

- Fold 1: RMSE = 0.8951, MAE = 0.6903
- Fold 2: RMSE = 0.9025, MAE = 0.6952
- Fold 3: RMSE = 0.8872, MAE = 0.6831
- Fold 4: RMSE = 0.8984, MAE = 0.6907
- Fold 5: RMSE = 0.8997, MAE = 0.6924

Ao final da validação cruzada, as médias das métricas foram calculadas:

- Média do RMSE: 0.8997
- Média do MAE: 0.6924

A média do RMSE de aproximadamente 0.8997 indica que, em média, as previsões do modelo diferem dos valores reais em cerca de 0.9 em uma escala de avaliação que vai de 1 a 5. Esse valor sugere que o modelo é capaz de fazer recomendações com um nível de precisão satisfatório.

Além disso, o MAE médio de 0.6924 reforça essa análise, mostrando que o erro absoluto médio das previsões é de aproximadamente 0.69. Em termos práticos, isso significa que as previsões do modelo, em média, estão a menos de 1 ponto de distância das avaliações reais fornecidas pelos usuários.

Esses resultados indicam que o modelo SVD utilizado é eficaz para a tarefa de recomendação de itens, apresentando erros relativamente baixos, o que o torna uma boa opção para sistemas de recomendação baseados em dados de avaliação de usuários.

3 REFERÊNCIAS

KAGGLE. The Movies Dataset. Disponível em: <https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?resource=download&select=movies_metadata.csv>. Acesso em: 30 set. 2024.

THE MOVIE DATABASE (TMDB) API. Disponível em: <https://developer.themoviedb.org/reference/intro/getting-started>. Acesso em: 30 set. 2024.

THE MOVIE DATABASE (TMDB). Disponível em: <https://www.themoviedb.org/>. Acesso em: 30 set. 2024.

PANDAS. Disponível em: <https://pandas.pydata.org/\>. Acesso em: 18 set. 2024.

NUMPY. Disponível em: <https://numpy.org/\>. Acesso em: 18 set. 2024.

MATPLOTLIB. Disponível em: <https://matplotlib.org/stable/contents.html\>. Acesso em: 18 set. 2024.

SEABORN. Disponível em: <https://seaborn.pydata.org/\>. Acesso em: 18 set. 2024.

WORDCLOUD. Disponível em: <https://pypi.org/project/wordcloud/\>. Acesso em: 18 set. 2024.

SURPRISE. Disponível em: <https://surpriselib.com/\>. Acesso em: 18 set. 2024.

SCIKIT-LEARN. Disponível em: <https://scikit-learn.org/stable/\>. Acesso em: 18 set. 2024.

PREVISÃO DE AVALIAÇÕES EM SISTEMAS DE RECOMENDAÇÃO PARA NICHOS DE MERCADO. Disponível em: <https://www.cos.ufrj.br/uploadfile/1365598708.pdf>. Acesso em: 22 set. 2024.

ESTUDO COMPARATIVO DE ALGORITMOS DE SISTEMAS DE RECOMENDAÇÃO DE FILMES. Disponível em: <https://www.maxwell.vrac.puc-rio.br/57546/57546.PDF> Acesso em: 22 set. 2024.

NETFLIX USERS SPEND 18 MINUTES PICKING SOMETHING TO WATCH, STUDY FINDS. Disponível em: <https://www.thewrap.com/netflix-users-browse-for-programming-twice-as-long-as-cable-viewers-study-says/>. Acesso em: 05 out. 2024.